20 Basic Linux Commands for Data Science Beginners



1. ls

The ls command is used to display the list of all the files and folders in the current directory.

```
$ ls
```

Output

```
AutoXGB_tutorial.ipynb  binary_classification.csv      requirements.txt
Images/                 binary_classification.csv.dvc  test-api.ipynb
LICENSE                 output/
README.md               output.dvc
```

2. pwd

It will display the full path of the current directory.

```
$ pwd
```

Output

```
C:\Repository\HuggingFace
```

3. cd

The cd command stands for change directory. By typing a new directory path, you can change the current directory. This command is essential for exploring the directory with multiple folders.

```
$ cd C:/Repository/GitHub/
```

```
HuggingFace  cd ../DagsHub
DAGsHub  cd C:\Repository\GitHub\
GitHub
```

4. wget

The wget allows you to download any file from the internet. In data science, it is use for downloading the data from data repositories.

```
$ wget https://raw.githubusercontent.com/
```

```
uiuc-cse/data-fa14/gh-pages/data/iris.csv
```

4. wget

Output

```
--2022-05-24 06:37:54--  https://raw.githubusercontent.com/uiuc-cse/data-fa14/gh-
pages/data/iris.csv
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.111.133,
185.199.109.133, 185.199.108.133, ...
Connecting to raw.githubusercontent.com
(raw.githubusercontent.com)|185.199.111.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3716 (3.6K) [text/plain]
Saving to: ‘iris.csv’

iris.csv            100%[===================>]   3.63K  --.-KB/s    in 0s

2022-05-24 06:37:55 (51.9 MB/s) - ‘iris.csv’ saved [3716/3716]
```

5. cat

Cat (concatenate) is a frequently used command to create, connect, and view files. The cat command reads the CSV file and displays the file content as output.

5. cat

```
$ cat iris.csv
```

Output

```
sepal_length,sepal_width,petal_length,petal_width,species
5.1,3.5,1.4,0.2,setosa
4.9,3,1.4,0.2,setosa
4.7,3.2,1.3,0.2,setosa
4.6,3.1,1.5,0.2,setosa
5,3.6,1.4,0.2,setosa
…………………………..
```

6. wc

wc (word count) is used to get information about word count, character count, and lines. In our case, it displays 4 columns as an output. The first column is line count, the second is word count, the third is character count, and the fourth is a file name.

```
$ wc iris.csv
```

Output

```
151  151 3716 iris.csv
```

7. head

The head command shows the top n lines in a file. In our case, it is displaying the top 5 lines in the iris.csv file.

```
$ head -n 5 iris.csv
```

Output

```
sepal_length,sepal_width,petal_length,petal_width,species
5.1,3.5,1.4,0.2,setosa
4.9,3,1.4,0.2,setosa
4.7,3.2,1.3,0.2,setosa
4.6,3.1,1.5,0.2,setosa
```

8. find

The find command is used to find files and folders, and by using `-exec`, you can execute other Linux commands on files and folders. In our case, we are finding all the files with ".dvc" extension.

```
$ find . -name "*.dvc" -type f
```

Output

```
./binary_classification.csv.dvc

./output.dvc
```

9. grep

It is used for filtering a particular pattern and displaying all the lines containing that pattern.

We are finding all the lines that contain “vir” in iris.csv

```
$ grep -i "vir" iris.csv
```

9. grep

```
grep -i "vir" iris.csv

>>> 6.3,3.3,6,2.5,virginica
>>> 5.8,2.7,5.1,1.9,virginica
>>> 7.1,3,5.9,2.1,virginica
>>> ...
>>> ...
```

10. history

History will show the log of the past commands. We have limited the output to display the 5 most recent commands.

```
$ history 5
```

Output

```
494   cat iris.csv
495   wc iris.csv
496   head -n 5 iris.csv
497   find . -name "*.dvc" -type f
498   grep -i "vir" iris.csv
```

11. zip

zip is used to compress the file size and file package utility. The first argument in the zip command is a zip file name, and the second is a file name or list of file names. The zip command is primarily used to compress and package datasets.

```
$ zip ZipFile.zip File1.txt File2.txt
```

12. unzip

It unzips or uncompresses the files and folders. Just provide a `.zip` file name, and it will extract all the files and folders in the current directory.

```
$ unzip sampleZipFile.zip
```

13. cp

It lets you copy a file, list of files, or directory to the destination directory. The first argument in the cp command is a file, and the second argument is the destination directory path.

```
$ cp a.txt work
```

14. mv

Similar to cp, the mv command lets you move a file, list of files, or a directory to another place. It is also used for renaming files and directories. The first argument in the mv command is a file, and the second is the path of destination directory.

```
$ mv a.txt work
```

15. rm

It removes files and directories from the file system. You can add a file or list of files names after the rm command.

```
$ rm b.txt c.txt
```

16. mkdir

It lets you create a directory of multiple directories at once. Just write the folder path after the mkdir command.

```
$ mkdir /love
```

Note: The user must have permission to create a folder in the parent directory.

17. rmdir

You can remove a directory or multiple directories by using rmdir. Just add a folder named as the first argument.

Note: The `-v` flag indicates verbose.

```
$ rmdir -v /love
```

```
VERBOSE: Performing the operation "Remove Directory" on target "C:\love".
```

18. man

It is used to display the manual of any command in the Linux system. In our case, we are going to learn about the echo command.

```
$ man echo
```

19. diff

It is used to display line-by-line differences between two files. Just add both files after the diff command to see the comparison.

```
$ diff app1.py app2.py
```

Output

```
31c31
<     solar_irradiation = loaded_model.predict(data)[1]

---

>     solar_irradiation = loaded_model.predict(data)[0]
```

20. alias

An alias is a productivity tool. I have shortened all your long and repetitive commands. I have shortened all of my Linux and Git commands to avoid making mistakes while writing the same command.

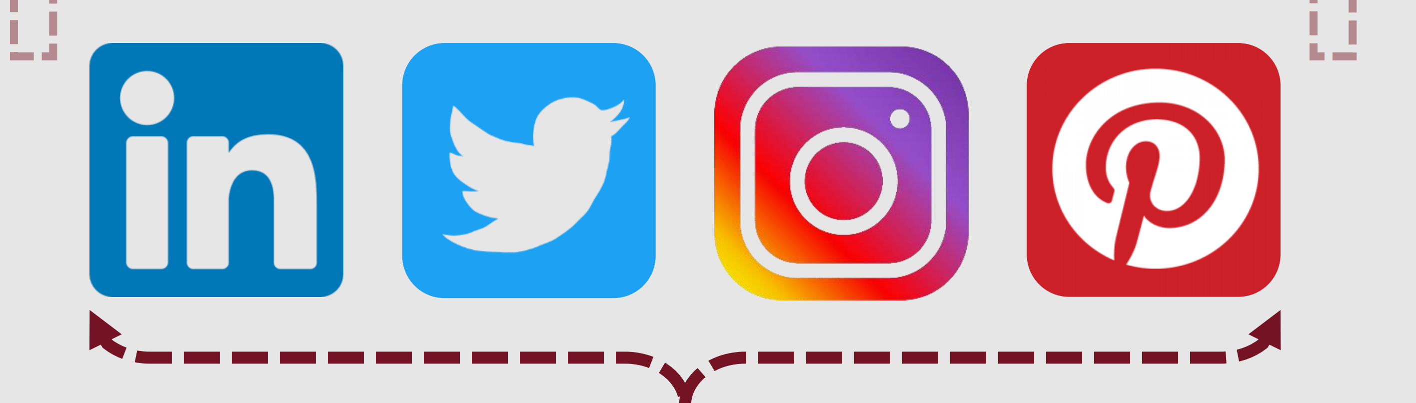
In the example below, the terminal is displaying the text “i love you” whenever I run the love command.

20. alias

```
$ alias love="echo 'i love you'"
```

```
alias love="echo 'i love you'"
love
>>> i love you
```

Stay in touch !



@DataCleanic

www.DataCleanic.ml